# Hide and Seek

Factors $N = UV$ in $O(N^{1/3} \log N)$ time probably

$O(N^{1/3+\varepsilon})$ time provably.

Survey

- trial division $O(\sqrt{N})$
- fastest provable algorithm:

  Pollard - Strassen $O(N^{1/4} \log^2 N)$

so hide and seek isn't so bad.

Other exponential time algorithms

- Lehman method $O(N^{1/3+\varepsilon})$, provably (actually $O(N^{1/3} \log\log$
- Pollard rho $O(\sqrt{p \log p})$ probably where $p$ is the smallest prime factor of $N$
- Shanks $O(N^{1/5+\varepsilon})$, assuming GRH
- Pollard p-1 method.

Some subexponential time algorithms

- Pomerance Quadratic Sieve

$$\exp\left((1+\varepsilon)\sqrt{\log N \log\log N}\right), \text{ for any } \varepsilon > 0.$$

- Number field sieve

$$\exp\left(\lambda (\log N)^{1/3} (\log\log N)^{2/3}\right), \text{ for any } \lambda > (32/9)^{1/3}$$

- Lenstra elliptic curve method

$$\exp\left((2+\varepsilon)\sqrt{\log p \log\log p}\right)$$

These all grew out of exponential time algorithms

ex Quadratic sieve tries to find solns to $x^2 = y^2 \bmod N$. Inspired by Lehman method.

Elliptic curve method inspired by Pollard $p-1$ method

All these algorithms (besides trial division) involve arithmetic mod $N$ or taking gcd's with $N$.

My goal was to go back to trial division and try to exploit info that trial division throws away - the remainder.

Another thing that strikes me is that all factoring algorithms can be described in just a few sentences. Mine is no exception. Proving that it works in claimed time is another issue.

To illustrate how easy it is to describe factoring algorithms, I'll describe Pollard-rho

Take any $x_0 \in \mathbb{Z}$

Let $x_{m+1} = x_m^2 + 1 \mod N$

Only finitely many possibilities, must eventually repeat. Same is true mod any $p | N$.

Say $x_{m'} = x_m \mod p$

Gives a 'random' sequence mod p that is also periodic from some $x_m$ onwards.

Birthday problem suggests $O(\sqrt{p \log p})$ steps needed to guarantee repetition.

Now $p | x_{m'} - x_m$ so $p | \gcd(x_{m'} - x_m, N)$

Odds are $N \nmid \gcd(x_{m'} - x_m, N)$ so gives non-trivial factor of N.

You might think you need to compare all $x_i - x_j$.
But no! In practice, one compares!!

$x_2 - x_1,\ x_4 - x_2,\ x_6 - x_3, \ldots$

Hide and Seek

Assume for simplicity that

$$N = UV \quad \text{with } U < V < 2U$$

so that $V^2 < 2N$, i.e $V < (2N)^{1/2}$

Let $a = \lceil (2N)^{1/3} \rceil > V^{2/3}$

Write

$$U = u_1 a + u_0$$

$$V = v_1 a + v_0$$

$$a > V^{2/3} \text{ so } \boxed{u_1, v_1} < V^{1/3} \boxed{< a^{1/2}}$$

Goal: determine $u_0, v_0, u_1, v_1$ and hence $U, V$.

Now

$$N = UV = u_0 v_0 \mod a$$

Not enough to determine $u_0, v_0$ uniquely
$\phi(a)$ possible solutions, one for every invertible congruence class mod $a$.

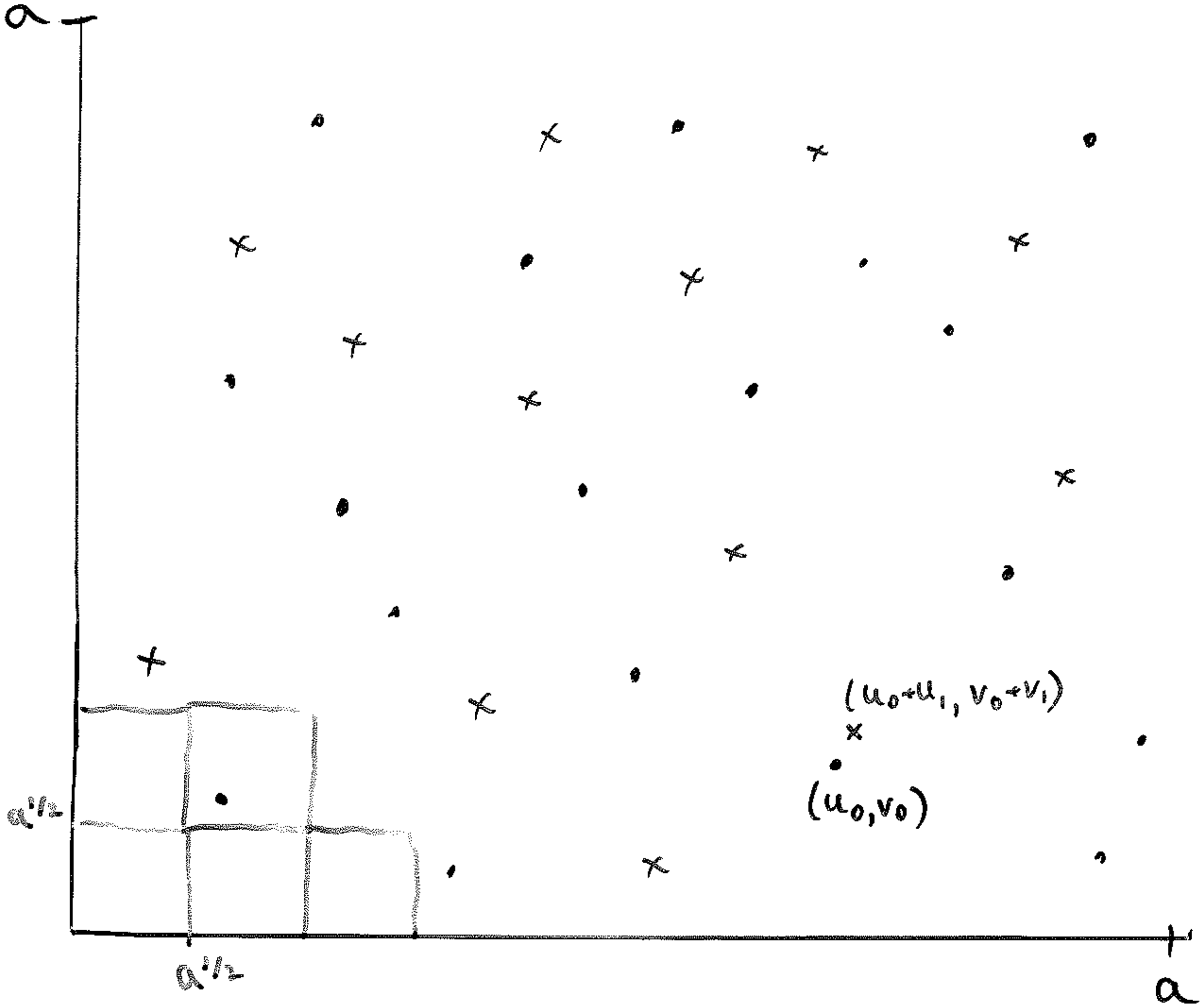
<u>But</u>

$$N = UV = (u_1 + u_0)(v_1 + v_0) \mod a-1$$

and $u_1, v_1$ are relatively small ($< a^{1/2}$)

so

$(u_0, v_0)$ and $(u_0+u_1, v_0+v_1)$ are close to one another. They give each other away by hiding near one another.




Compute all solns to $xy = N \bmod a$, denoted by •'s
and to $xy = N \bmod a-1$, denoted by x's

takes $\Theta(a)$ time.

Throw into $a \times a$ square which is partitioned into smaller $a^{1/2} \times a^{1/2}$ squares ($\sim a$ of them)

expect # pts per □ $= \Theta(1)$

$(u_0, v_0)$ and $(u_0+u_1, v_0+v_1)$ fall in same □ or in two neighbouring □'s.

Any 2 pts gives us candidate $(u_0, v_0)$ and $(u_0+u_1, v_0+v_1)$, i.e candidate $U$ and $V$ and we check if they give $N = UV$.

time: $O(a)$ to compute all solns to $xy \equiv N \bmod a$ and to $xy \equiv N \bmod a-1$

· expect $O(a)$ time to scan across all little □'s and their immediate neighbours comparing pairs of points.

$a = O(N^{1/3})$, so gives $O(N^{1/3})$ predicted running time.

I can prove $O(N^{1/3+\varepsilon})$ for any $\varepsilon > 0$.

## Variant

In general no restriction on $U$ and $V$

other than $1 < U < V < N$

Let $U = N^{\alpha}$, $V = N^{1-\alpha}$, $\alpha \leq 1/2$

Take $a \approx N^{1/3}$ and instead of $\square$'s of size $a^{1/2} \times a^{1/2}$, use rectangles of size $w \times h$ where $w \approx N^{\alpha - 1/3}$, $h \approx N^{1-\alpha-1/3}$.

Assume $\alpha > 1/3$ (if not we can find $U$ in $O(N^{1/3})$ time by trial division).

Area of each rectangle $\approx N^{1/3}$
of $a \times a$ square $\approx N^{2/3}$

so $O(N^{1/3})$ rectangles, and again we expect $O(1)$ pts per rectangle, so $O(N^{1/3})$ running time.

<u>However</u>, since we don't know $U, V$ apriori we start with $w$ small, and keep doubling it until $w > N^{\alpha - 1/3}$ and set $h = \frac{N^{1/3}}{w}$ and we factor $N$.
Overall running time is thus predicted to be $O(N^{1/3} \log N)$

Towards subexponential

Decrease $a$, fewer points.

Problem: $u_1, v_1$ grow, no longer small.

Solution? use polynomial expansion instead?

For simplicity assume $a^d \le u, v < a^{d+1}$

write: $$u = \sum_0^d u_j a^j \qquad 0 \le u_j, v_j < a$$

$$v = \sum_0^d v_j a^j$$

Then

$$N = uv$$

$$= \underbrace{\sum_0^d u_j \delta^j}_{\text{denote } u(\delta)} \underbrace{\sum_0^d v_j \delta^j}_{v(\delta)} \mod a - \delta$$

We'll need $d+1$ pts to uniquely determine the coefficients $u_j, v_j$. If we take $d+2$ pts we'll typically overdetermine the system unless the points genuinely come from $u(\delta), v(\delta)$.

Now

$$u(\delta),\ v(\delta) < a\left(\delta^d + \dots \delta + 1\right)$$

For each $\delta = 0, 1, 2, \dots d+1$ list all $(x, y)$ such that:

$$xy = N \bmod a - \delta$$

$$0 < x, y < a\left(\delta^d + \dots \delta + 1\right)$$

assume as before $\gcd(a-\delta, N) = 1$ otherwise we easily get a factor of $N$.

$$\#\text{ of pts}: \ \phi(a-\delta)\left(\delta^d + \dots \delta + 1\right)^2$$

$$= O\left(a d^{2d}\right)$$

Need a method to find

$$(u(\delta), v(\delta)), \quad \delta = 0, 1, 2, \ldots$$

hiding amongst all the $(x, y)$'s.

## Question

Let $X > 0$,

$S_0, S_1, \ldots S_{d+1}$ $d+2$ sets of points $\in \mathbb{Z}^2$ all of whose coordinates are $> 0$ and $\leq X$.

Assume that amongst these, there exists $d+2$ pts, one from each $S_\delta$, whose coordinates are given by $u(\delta), v(\delta) \in \mathbb{Z}[\delta]$,

i.e for each $\delta = 0, 1, \ldots d+1$ there exists $(x_\delta, y_\delta) \in S_\delta$ such that

$$x_\delta = u(\delta), \quad y_\delta = v(\delta).$$

Can one find these $d+2$ pts more efficiently than by exhaustively searching through all possible $d+2$ tuples, for example in $O(X^\alpha d^{\beta d})$ time, for some $\alpha, \beta > 0$?

In our application

$$X = O(ad^{2d}).$$

Also $N = uv$, $a^d \leq u < v < a^{d+1}$

so $a < N^{1/2d}$.

Taking $d$ proportionate to

$$\left(\frac{\log N}{\log\log N}\right)^{1/2}$$

would give a factoring algorithm requiring

$$\exp\left(\gamma\left(\log N \log\log N\right)^{1/2}\right)$$

time and storage, for some $\gamma > 0$.

Uniform distribution of solns $(x,y)$ to $xy \equiv N \bmod a$.

Let $\begin{matrix} 0 \le x_1 < x_2 < a \\ 0 \le y_1 < y_2 < a \end{matrix}$, $x_1, x_2, y_1, y_2 \in \mathbb{Z}$

$R$, rectangle

$$R = R(x_1, x_2, y_1, y_2)$$

$$= \left\{ (x,y) \in \mathbb{Z}^2 \;\middle|\; \begin{matrix} x_1 \le x < x_2 \\ y_1 \le y < y_2 \end{matrix} \right\}$$

Let $C_R(N,a)$ denote the number of solns to $xy \equiv N \bmod a$ that lie in $R$.

$$C_R(N,a) = \sum_{\substack{(x,y) \in R \\ xy \equiv N \bmod a}} 1$$

Thm

$$C_R(N,a) = \frac{\text{area}(R)}{a^2}\,\phi(a) + O\left(a^{1/2+\varepsilon}\right)$$

for any $\varepsilon > 0$.

remark: shows $R$ gets its fair share of solns when area of $R$ is larger than $a^{3/2+\varepsilon}$, so if $R$ is $\square$, it should have side length $a^{3/4+\varepsilon}$.

proof can find in literature with $N=1$ or a prime. will outline since the techniques are also used to prove running time.

How to detect solns to $xy \equiv N \bmod a$

$$\frac{1}{a}\sum_{k=0}^{a-1} e\left(\frac{k}{a}(y-\bar{x}N)\right) = \begin{cases} 1 & \text{if } xy \equiv N \bmod a \\ 0 & \text{otherwise} \end{cases}$$

notation: $e(z) = \exp(2\pi i z)$

$\bar{x} = x^{-1} \bmod a$.

---

So

$$C_R(N,a) = \frac{1}{a}\sum_{k=0}^{a-1} \sum_{\substack{(x,y)\in R \\ \gcd(x,a)=1}} e\left(\frac{k}{a}(y-\bar{x}N)\right)$$

$k=0$ gives the main contribution

<u>lemma</u>

$k=0$ term equals

$$\frac{\text{area}(R)}{a^2}\phi(a) + O(a^{\varepsilon})$$

## $k \geq 1$ terms

Lemma: for any $\varepsilon > 0$

$$\frac{1}{a} \sum_{k=1}^{a-1} \sum_{\substack{(x,y) \in R \\ \gcd(x,a)=1}} e\left(\frac{k}{a}(y - \bar{x}N)\right) = O(a^{1/2+\varepsilon})$$

pf

separate the $y$ sum, gives geometric series

$$\frac{e\left(\frac{k}{a}y_2\right) - e\left(\frac{k}{a}y_1\right)}{e\left(\frac{k}{a}\right) - 1}$$

take absolute values:

$$\frac{1}{a} \sum_{k=1}^{a-1} \left| \frac{\sin \frac{\pi k}{a}(y_2 - y_1)}{\sin \frac{\pi k}{a}} \right| \left| \sum_{\substack{x_1 \leq x < x_2 \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x}N\right) \right|$$

Group $k, a-k$ terms together

Possible middle term (if $a-1$ is odd) contributes $O(1)$.

Use $\sin(t) \le \min(t, 1)$, $t \ge 0$

$1/\sin(t) < 2/t$, $0 < t < \pi/2$

$$\sum_{\substack{x_1 \le x < x_2 \\ \gcd(x,a)=1}} e\left(-\frac{k}{a}\bar{x}n\right) = O\left(a^{1/2+\varepsilon} \gcd(k,a)\right)$$

use discrete f.t. to write as a full sum involves Kloosterman sums, apply Weil bound.

Break the sum over k in two pieces

$$\left( k \le \frac{a}{\pi(y_2 - y_1)}, \text{ and } \frac{a}{\pi(y_2 - y_1)} < k \le \frac{a-1}{2} \right)$$

After some work one gets

$$O\left(a^{1/2+\varepsilon}\right).$$

$$\sum_{\substack{x_1 \le x < x_2 \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x} N\right) = \frac{1}{a} \sum_{\substack{0 \le x < a \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x} N\right) \sum_{\substack{x_1 \le m < x_2 \\ 0 \le n < a}} e\left(\frac{n}{a}(m-x)\right)$$

$$= \frac{1}{a} \sum_{n=0}^{a-1} \underbrace{S(-n,-kN,a)}_{\substack{\text{Kloosterman} \\ \text{sum}}} \underbrace{\sum_{x_1 \le m < x_2} e\left(\frac{mn}{a}\right)}_{\substack{\text{geometric series, in} \\ \text{absolute value is} \\ \frac{\left|\sin\left(\frac{\pi n}{a}(x_2-x_1)\right)\right|}{\left|\sin\left(\frac{\pi n}{a}\right)\right|}}}$$

$$S(-n,-kN,a) = \sum_{\substack{0 \le x < a \\ \gcd(x,a)=1}} e\left(-\frac{nx + Nk\bar{x}}{a}\right)$$

Weil

$$|S(-m,-Nk,a)| \le \tau(a) \gcd(m,k,a)^{1/2} a^{1/2}$$

$$= O\left(a^{1/2+\varepsilon} \gcd(k,a)^{1/2}\right)$$

# Running time

Assume $U < V < 2U$, so we're partitioned $a \times a$ square into □'s of side length $a^{1/2}$.

Time to compare 2 □'s, say $S_1, S_2$:

$$O\left(C_{S_1}(N,a) \cdot C_{S_2}(N,a-1)\right)$$

$$= O\left(C_{S_1}(N,a)^2 + C_{S_2}(N,a)^2\right)$$

So we need to bound 2nd moments

$$\sum_S C_S(N,a)^2 \quad \text{and} \quad \sum_S C_S(N,a-1)^2$$

Make a small adjustment to □'s.

Take, for 1st sum, □'s of sidelength $b = \lceil a^{1/2} \rceil$, and then $b \to b+1$ until $\gcd(b,a) = 1$. $b = a^{1/2} + O(a^{\varepsilon})$.

For 2nd sum use $b = \lceil (a-1)^{1/2} \rceil$ etc.

Prefer not to truncate abrubtly, so take $ab \times ab$ square instead of $a \times a$ square.

Let

$$B_{ij} = \left\{ (x,y) \in \mathbb{Z}^2 \;\middle|\; \begin{array}{l} ib \le x < (i+1)b \\ jb \le y < (j+1)b \end{array} \right\}$$

$$b^2 \sum_{B \in a \times a \text{ square}} C_B(N,a)^2 = O\left( \sum_{B \in ab \times ab \text{ square}} C_B(N,a)^2 \right)$$

But

$$C_B(N,a)^2 = \frac{1}{a^2} \sum_{0 \le k_1, k_2 \le a-1} \sum_{\substack{(x_1,y_1) \in B \\ (x_2,y_2) \in B \\ \gcd(x_1 x_2, a) = 1}} e\left( \frac{k_1}{a}(y_1 - \bar{x}_1 N) - \frac{k_2}{a}(y_2 - \bar{x}_2 N) \right)$$

sum over $y_1, y_2$ is a product of two geometric series.

Summing over $B \in ab \times ab$ square

i.e over $B_{ij}$ with $0 \le i, j \le a-1$ gives

(only the terms with $k_1 = k_2$ contribute)

$$\sum_{B \in ab \times ab \text{ square}} C_B(N,a)^2$$

$$= \frac{1}{a} \sum_{k=0}^{a-1} \left( \sum_{i=0}^{a-1} \sum_{\substack{ib \le x_1, x_2 < (i+1)b \\ \gcd(x_1 x_2, a) = 1}} e\left(-\frac{N}{a} k (\bar{x}_1 - \bar{x}_2)\right) \right) \left| \frac{e\left(\frac{kb}{a}\right) - 1}{e\left(\frac{k}{a}\right) - 1} \right|^2$$

$k=0$ contributes $O(\phi(a)^2)$.

for other terms, $1 \le k \le a-1$, write the sum over $i$ in terms of Kloosterman sums.

Let

$$A_{x_1, x_2}(t) = \begin{cases} 0 & \text{if } \gcd(x_1 x_2, a) > 1 \\ e\left(\frac{t}{a}(\bar{x}_1 - \bar{x}_2)\right) & \text{otherwise} \end{cases}$$

The sum over $i$ equals:

$$\sum_{i=0}^{a-1} \sum_{ib \le x_1, x_2 < (i+1)b} A_{x_1,x_2}(-Nk)$$

Use 2-dim discrete fourier transform

$$\hat{A}_{m_1,m_2}(t) = \sum_{0 \le x_1, x_2 \le a-1} A_{x_1,x_2}(t)\, e\left(-\frac{m_1x_1+m_2x_2}{a}\right)$$

so that

$$A_{x_1,x_2}(t) = \frac{1}{a^2} \sum_{0 \le m_1, m_2 \le a-1} \hat{A}_{m_1,m_2}(t)\, e\left(\frac{m_1x_1+m_2x_2}{a}\right)$$

Hence the $i$ sum equals

$$\frac{1}{a^2} \sum_{0 \le m_1, m_2 \le a-1} \hat{A}_{m_1,m_2}(-Nk) \underbrace{\left(\sum_{i=0}^{a-1} \sum_{ib \le x_1, x_2 < (i+1)b} e\left(\frac{m_1x_1+m_2x_2}{a}\right)\right)}_{\substack{\text{product geometric series;}\\ \text{only get contribution}\\ \text{if } m_1 \equiv -m_2 \bmod a}}$$

So, i sum equals

$$\frac{1}{a}\sum_{m=0}^{a-1} \hat{A}_{m,a-m}(-Nk)\left|\frac{e(\frac{mb}{a})-1}{e(\frac{m}{a})-1}\right|^2$$

But

$$\hat{A}_{m,a-m}(-Nk) = \sum_{\substack{0\le x_1,x_2\le a-1\\ \gcd(x_1x_2,a)=1}} e\left(-\frac{Nk}{a}(\bar{x}_1-\bar{x}_2)\right)e\left(-\frac{(mx_1-mx_2)}{a}\right)$$

$$=\left|\sum_{\substack{0\le x\le a-1\\ \gcd(x,a)=1}} e\left(-\frac{(Nk\bar{x}+mx)}{a}\right)\right|^2 = |S(-m,-Nk,a)|^2$$

Kloosterman sum

Thus

$$\sum_{B \in ab \times ab \text{ square}} C_B(N,a)^2$$

identity.

$$= \frac{1}{a^2} \sum_{k=0}^{a-1} \sum_{m=0}^{a-1} \left| S(-m,-Nk,a) \right|^2 \left| \frac{e\left(\frac{mb}{a}\right)-1}{e\left(\frac{m}{a}\right)-1} \right|^2 \left| \frac{e\left(\frac{kb}{a}\right)-1}{e\left(\frac{k}{a}\right)-1} \right|^2$$

Weil

$$\left| S(-m,-Nk,a) \right| \leq \tau(a) \gcd(m,k,a)^{1/2} a^{1/2}$$

$$= O\left(a^{1/2+\varepsilon} \gcd(k,a)^{1/2}\right)$$

Pull out $k=0$ contribution, $O(\phi(a)^2)$

For other terms, apply Weil bound, separate sums, proceed as before to get, for the above,

$$O\left(b^2 a^{1+\varepsilon}\right)$$

and so, dividing by $b^2$ gives

$$\sum_{B \in a \times a \text{ square}} C_B(N,a)^2 = O\left(a^{1+\varepsilon}\right).$$